Rapport projet BI :

1. Présentation du projet :

Ce projet s'inscrit dans une démarche de Business Intelligence (BI) appliquée à la base de données **Northwind**, une entreprise fictive spécialisée dans l'import-export des produits. L'objectif principal est de transformer des données brutes stockées dans un système de gestion de base de données relationnelle (SQL Server) en un outil visuel d'aide à la décision.

2. Problématique :

Les gestionnaires de Northwind font face à un volume de données croissant et ont besoin de répondre rapidement à des questions stratégiques :

1. Quels sont les produits les plus rentables ?
3. Quels employés réalisent les meilleures performances commerciales ?
4. Comment se porte la logistique en termes de délais et de volumes de livraison ?

3.Objectifs du Dashboard :

- Suivre l'évolution du chiffre d'affaires en temps réel.
- Identifier les zones géographiques les plus actives.
- Analyser l'efficacité de la chaîne logistique.

4.Environnement Technique :

Le projet a été réalisé en utilisant une stack technologique moderne :

- **SQL Server** : Pour le stockage et l'extraction initiale des données.
- **Python (Pandas)** : Pour le nettoyage (ETL) et la structuration des données.
- **Dash & Plotly** : Pour la création de l'interface utilisateur et des visualisations graphiques.
- **Git & GitHub** : Pour le versionnage du code et la collaboration.

5.Justification de l'extraction directe via SQLAlchemy :

Bien qu'un fichier Excel ait été mis à disposition dans le cadre du projet, le choix technique a été fait de ne pas l'utiliser comme source de données. En Business Intelligence, l'exploitation directe d'une base de données relationnelle permet une meilleure compréhension du processus d'extraction et se rapproche davantage des pratiques professionnelles.
Ainsi, les données ont été extraites directement depuis le serveur SQL Server à l'aide de la bibliothèque Python SQLAlchemy. Ce choix est justifié par les raisons suivantes :

1. Fiabilité et intégrité des données :

L'extraction directe depuis la base SQL Server garantit l'utilisation de la source de données officielle, sans risque d'altérations manuelles pouvant survenir lors de la manipulation d'un fichier Excel.

2. Maîtrise complète du processus ETL :
Cette approche permet de mettre en œuvre une véritable étape d'Extraction (E) en configurant explicitement la connexion à la base de données, ce qui n'aurait pas été le cas avec un simple import de fichier plat.
3. Automatisation et évolutivité :
La connexion directe entre Python et SQL Server rend le processus reproductible et permet une mise à jour automatique des analyses et du tableau de bord lorsque les données de la base évoluent, contrairement à un fichier Excel statique.

4.Python et Pandas :

« Le choix de **Python**, et particulièrement de la bibliothèque **Pandas**, a été privilégié pour ce projet car il représente le standard absolu dans le domaine de la Data Science et de la Business Intelligence moderne.
Bien que d'autres outils de manipulation de données existent, Python offre une flexibilité que les outils classiques ne permettent pas. L'utilisation de **Pandas** a permis de transformer des données brutes et complexes issues de SQL en structures de données organisées (DataFrames) avec une efficacité chirurgicale. Ce choix est justifié par la nécessité d'automatiser le nettoyage des données : là où un traitement manuel est long et risqué, un script Python garantit que chaque transformation (calcul de marge, gestion des dates, fusion de tables) est reproductible à l'infini sur de nouvelles données sans aucun effort supplémentaire. Utiliser Python, c'est choisir la puissance de calcul et la rigueur de l'automatisation. »

6.Description du processus de réalisation du projet :

Interconnexion avec la base de données SQL Server

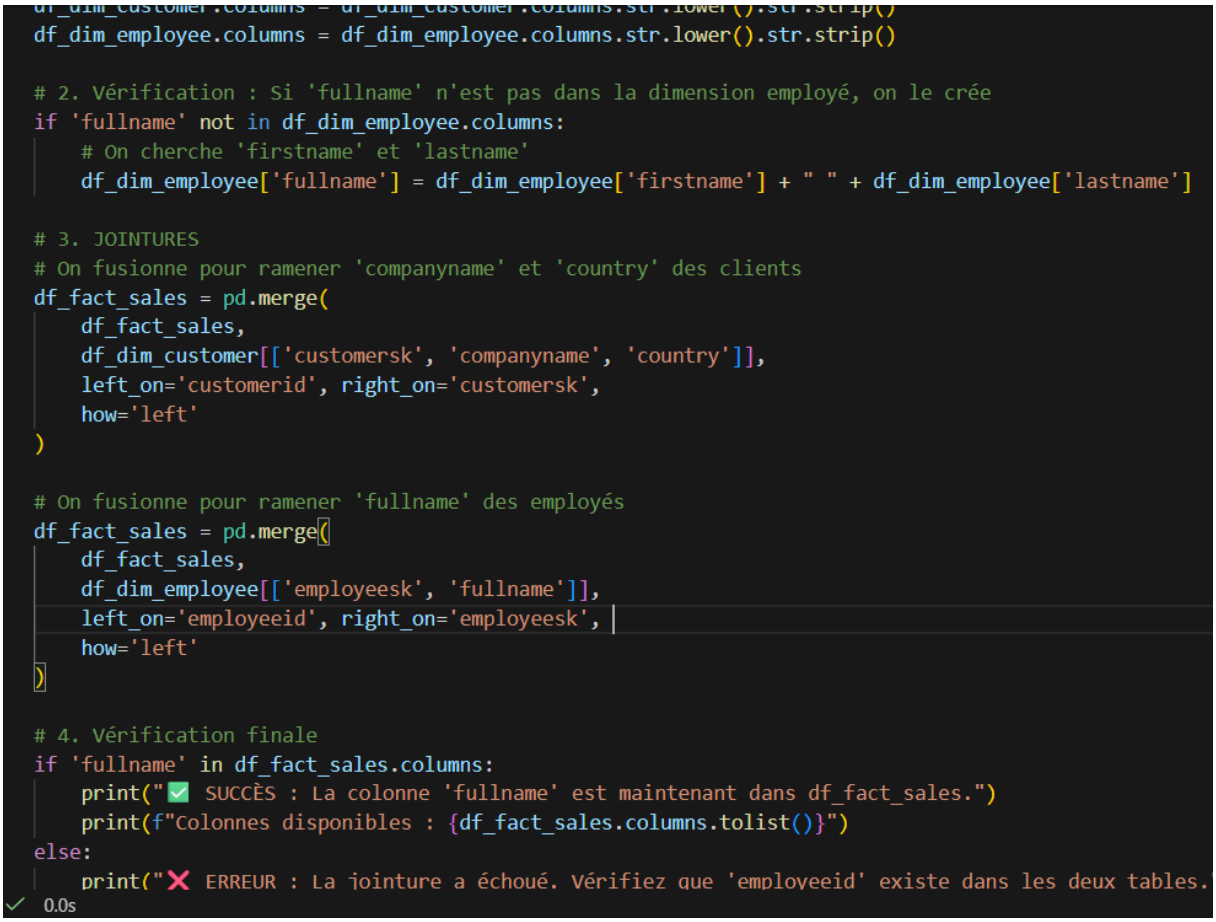
Le projet a débuté par l'établissement d'un pont de communication entre l'environnement de développement Python et le serveur **SQL Server**. En utilisant la bibliothèque **SQLAlchemy**, j'ai configuré un moteur de connexion capable d'interroger directement la base de données **Northwind**. Cette approche a permis d'extraire les tables relationnelles nécessaires (notamment les commandes, les détails de commandes et les employés) tout en garantissant l'intégrité et la sécurité des données sources.

Extraction et ingestion des données via Pandas

Une fois la connexion établie, les données brutes ont été importées dans des structures **DataFrames** via la bibliothèque **Pandas**. Ce passage du format SQL au format Python est essentiel car il offre une puissance de manipulation supérieure. À ce stade, j'ai pu visualiser l'état initial des données et identifier les besoins de nettoyage, tels que le formatage des dates ou la gestion des valeurs manquantes dans les registres d'expédition.

Transformation et enrichissement du jeu de données (ETL)

Cette phase constitue le cœur technique du projet. J'ai procédé à la consolidation des informations en effectuant des jointures (merges) entre les différentes tables extraites. L'objectif était de créer une table unique et cohérente où chaque ligne de commande est associée à son vendeur et à son chiffre d'affaires. Pour cela, j'ai implémenté une logique de calcul pour générer le montant total de chaque transaction (Prix Quantité Remise), transformant ainsi des données transactionnelles éparpillées en indicateurs de performance (KPI) exploitables.



Persistance du jeu de données final

Après avoir finalisé les transformations, j'ai exporté le résultat nettoyé vers un fichier **CSV** nommé df_final_data.csv. Cette étape de chargement (Loading) permet de séparer la préparation technique de la visualisation. En stockant les données transformées, on évite de solliciter inutilement le serveur SQL à chaque consultation du dashboard, ce qui

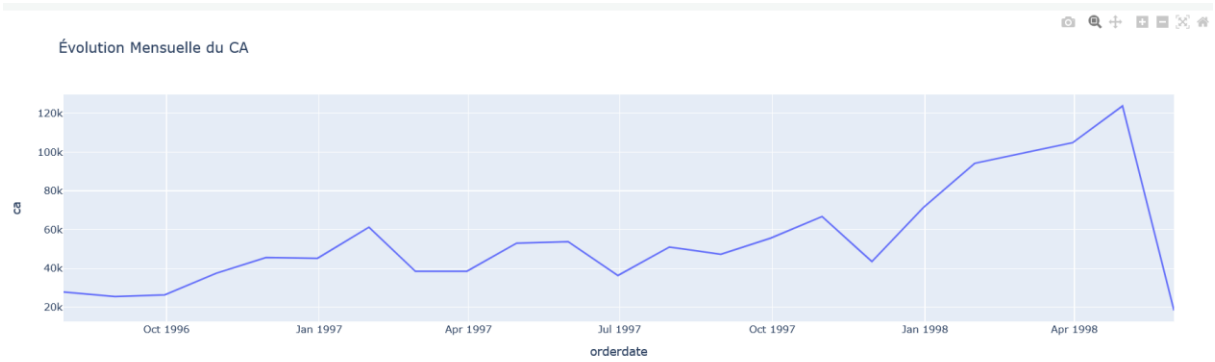
optimise les performances de l'application finale.

```python
import plotly.express as px
import plotly.io as pio

# On force l'ouverture dans le navigateur pour éviter les erreurs de rendu VS Code
pio.renderers.default = "browser"

# 1. Top 5 Employés
df_top_emp = df_fact_sales.groupby('fullname')['ca'].sum().nlargest(5).reset_index()

# 2. Top 5 Clients
df_top_cust = df_fact_sales.groupby('companyname')['ca'].sum().nlargest(5).reset_index()

# --- GRAPHIQUES ---

fig_emp = px.bar(df_top_emp, x='fullname', y='ca', color='ca',
                 title="Top 5 Employés par CA", color_continuous_scale='GnBu')

fig_cust = px.bar(df_top_cust, x='companyname', y='ca', color='ca',
                  title="Top 5 Clients par CA", color_continuous_scale='Viridis')

fig_emp.show()
fig_cust.show()

# Export pour le Dashboard Dash
df_fact_sales.to_csv("../df_final_data.csv", index=False)
```

✓ 4.5s

Conception et déploiement du Dashboard Dash/Plotly

La dernière étape a consisté à développer l'interface utilisateur interactive. En utilisant le framework **Dash**, j'ai structuré une mise en page web intégrant des graphiques dynamiques générés par **Plotly**. J'ai programmé des fonctions de filtrage (callbacks) permettant aux utilisateurs de segmenter les performances par employé ou par pays en temps réel. Le résultat final est un outil d'aide à la décision qui traduit des lignes de code complexes en visualisations claires et interactives.



7. Indicateurs clés de performance (KPI) :

Les KPI suivants ont été intégrés au projet pour permettre un pilotage précis de l'activité de Northwind :

- **Chiffre d'Affaires Total (Sales Amount) :** Calculé dans la **Cellule 1** , cet indicateur permet de mesurer la santé financière globale de l'entreprise sur la période.
- **Évolution Mensuelle des Ventes :** Généré dans la **Cellule 2** par agrégation temporelle, ce KPI permet d'identifier les saisonnalités et les tendances de croissance ou de baisse d'activité.
- **Performance Commerciale (Top 5 Employés) :** Extrait dans la **Cellule finale** via un groupement par fullname. Il permet d'identifier les vendeurs les plus performants en termes de génération de revenus.
- **Portefeuille Client (Top 5 Clients) :** Établi grâce à la jointure avec la table Customers, ce KPI identifie les clients stratégiques qui contribuent le plus au chiffre d'affaires.
- **Volume de Commandes :** Indicateur de base permettant de suivre le flux logistique et l'activité opérationnelle globale.

Conclusion :

Ce projet a permis de mettre en place une solution complète de Business Intelligence pour l'entreprise Northwind, couvrant l'intégralité du cycle de la donnée. À partir d'une extraction directe depuis SQL Server via SQLAlchemy, l'intégrité des informations a été préservée, tandis que les processus de nettoyage et de transformation ont été automatisés à l'aide de Pandas.

Le passage de données transactionnelles brutes à un dashboard interactif développé avec Dash et Plotly offre désormais une visibilité immédiate sur les performances commerciales et les tendances de ventes mensuelles. Cette approche moderne remplace avantageusement les manipulations manuelles sous Excel, en proposant un outil d'aide à la décision fiable, reproductible et évolutif.

Ce travail met en évidence ma capacité à mobiliser et orchestrer des outils techniques variés afin de répondre efficacement à des problématiques métiers concrètes, tout en respectant les principes fondamentaux de la Business Intelligence moderne.